

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

До захисту допущено
Завідувач кафедри

Сергій Стіренко

“ ____ ” _____ 2020 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Комп'ютерні системи та мережі»
спеціальності 123 «Комп'ютерна інженерія»
на тему: «Оповіщувальна комунікаційна система»

Виконав:

Студент IV курсу, групи ІО-64
Дмитро МАКСИМЧУК

Керівник:

Старший викладач
Юрій ВІНОГРАДОВ

Консультант з нормконтролю:

Професор кафедри ОТ, д.т.н.,
Валерій СІМОНЕНКО

Рецензент:

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ - 2020 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 123 «Комп'ютерна інженерія»
Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ
Завідувач кафедри

Сергій Стіренко
"___" _____ 2020 року

ЗАВДАННЯ
на дипломний проєкт студенту

Максимчуку Дмитру Анатолійовичу

1. Тема проєкту «Оповіщувальна комунікаційна система» керівник проєкту Виноградов Юрій Миколайович, затверджені наказом по університету від "07" травня 2020 року № 1081-с.
2. Термін подання студентом проєкту 26 травня 2020р.
3. Вихідні дані до проєкту див. технічне завдання .
4. Зміст пояснювальної записки Аналіз і характеристика об'єкта проєктування, обґрунтування оптимального варіанта реалізації мети цієї роботи, розробка додатку: вибір технологій та їх обґрунтування, основні рішення з реалізації додатку. Висновки.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) принципова схема, функціональна схема, структурна схема.
6. Консультанти розділів проєкту

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| нормконтроль | Сімоненко В.П., професор, д.т.н. | | |

7. Дата видачі завдання 01.09.2019.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту | Термін виконання етапів проєкту | Примітка |
|-------|--|---------------------------------|----------|
| 1 | <i>Затвердження теми роботи</i> | <i>01.09.2019</i> | |
| 2 | <i>Вивчення та аналіз завдання</i> | <i>15.12.2019-15.03.2020</i> | |
| 3 | <i>Розробка архітектури додатку</i> | <i>15.03.2020-25.03.2020</i> | |
| 4 | <i>Написання програмної частини</i> | <i>25.03.2020-05.04.2020</i> | |
| 5 | <i>Тестування та виправлення помилок</i> | <i>05.04.2020-15.04.2020</i> | |
| 6 | <i>Оформлення пояснювальної записки</i> | <i>15.04.2020-20.05.2020</i> | |
| 7 | <i>Захист програмного продукту</i> | <i>25.04.2020</i> | |
| 8 | <i>Передзахист</i> | <i>26.05.2020</i> | |
| 9 | <i>Захист</i> | <i>18.06.2020</i> | |

Студент

Дмитро МАКСИМЧУК

Керівник

Юрій ВІНОГРАДОВ

Анотація

Робота присвячена розробці оповіщувальної комунікаційної системи. Кількість інтернет-користувачів щоденно зростає, а тому з'являється можливість автоматизації та спрощення процесів життєдіяльності. Більшість повсякденних речей вже можна робити онлайн, адже це зручно, безпечно та надійно.

Мета оповіщувальної комунікаційної системи полягає у тому, щоб спростити процес привітання члена організації або компанії з днем народження чи будь-яким іншим святом. Програма автоматизує процедуру організації події та економить час як для організатора, так і для учасників. Завдяки цьому, зменшується вірогідність фактору людської помилки та з'являються нові можливості, які недоступні в реальному житті.

Онлайн система дає змогу імениннику вказати свої інтереси та подарунок. А організатор отримує можливість просто та завчасно зібрати кошти на подарунок.

Аннотация

Работа посвящена разработке уведомительной коммуникационной системы. Количество интернет-пользователей каждый день растёт, а потому появляется возможность автоматизации та упрощения процессов жизнедеятельности. Большинство ежедневных вещей уже можно делать онлайн, потому что это удобно, безопасно и надёжно.

Цель уведомительной коммуникационной системы состоит в том, чтобы упростить процесс поздравления члена организации или компании с днем рождения или любым другим праздником. Программа автоматизирует процедуру организации события и экономит время как для организатора, так и для участников. Благодаря этому, уменьшается вероятность фактора человеческой ошибки и появляются новые возможности, которые недоступны в реальной жизни.

Онлайн система дает возможность имениннику указать свои интересы и подарок. А организатор получает возможность просто и вовремя собрать средства на подарок.

Abstract

The work is devoted to the development of a notification communication system. The number of Internet users is growing every day, and therefore it becomes possible to automate and simplify the processes of life. Most daily things can already be done online, because it is comfortable, safe and reliable.

The purpose of a notification communication system is to simplify the process of congratulating a member of an organization or company on a birthday or any other holiday. The program automates the event organization procedure and saves time for both the organizer and participants. Due to this, the probability of the human error factor decreases and new opportunities appear that are not available in real life.

The online system allows the birthday party to indicate their interests and gift. And the organizer gets the opportunity to simply and timely raise funds for a gift.

Технічне завдання до дипломного проекту

ЗМІСТ

| | |
|---|---|
| 1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ | 2 |
| 2. ПІДСТАВИ ДЛЯ РОЗРОБКИ | 2 |
| 3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ..... | 2 |
| 4. ДЖЕРЕЛА РОЗРОБКИ..... | 2 |
| 5. ТЕХНІЧНІ ВИМОГИ..... | 2 |
| 5.1. Вимоги до розроблюваного продукту..... | 2 |
| 5.2. Вимоги до програмного забезпечення | 2 |
| 5.3. Вимоги до апаратного забезпечення | 3 |

| | | | | | | | | | |
|-----------|-----------------|----------|-------|-----|---|--|-------|---------|--|
| | | | | | ІАЛЦ.467100.002 ТЗ | | | | |
| | | | | | | | | | |
| Зм. | Арк. | № докум. | Підп. | Дат | | | | | |
| Розробив | Максимчук Д.А. | | | | Оповіщувальна комунікаційна система Технічне завдання | Літ. | Аркуш | Аркушів | |
| Перевір. | Виноградов Ю.М. | | | | | | 1 | 3 | |
| | | | | | | | | | |
| Н. контр. | Симоненко ВП. | | | | | НТУУ “КПІ імені Ігоря Сікорського”, ФІОТ Група ІО-64 | | | |
| Затверд. | | | | | | | | | |

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання розповсюджується на розробку оповіщувальної комунікаційної системи.

Область застосування: організації та компанії.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання розробки оповіщувальної комунікаційної системи, затвердженою кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проекту є розробка оповіщувальної комунікаційної системи для автоматизації організації привітань членів організацій та компаній.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами для розробки служать науково-технічна література з комп'ютерних технологій, довідники з програмування, публікації в Інтернеті за даним питанням.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розроблюваного продукту

- Розробка інтерфейсу для іменинника, організатора та учасників.
- Панель керування подією.
- Інформаційні блоки та оповіщення для учасників привітань.
- Система сповіщень.
- Інтеграція з базою організації чи компанії.
- Інтеграція з комунікаційною системою.

5.2. Вимоги до програмного забезпечення

- Операційна система MS Windows 7 і вище, або Linux 14 і вище.
- Python 3 і вище.
- JavaScript бібліотека React 14 і вище.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.002 ТЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 2 |

- Slack Workspace.
- Програмний інтерфейс (API) або база даних з інформацією про користувачів організації чи компанії.

5.3. Вимоги до апаратного забезпечення

- Комп'ютер на базі процесору Intel Core i3 і вище.
- Оперативної пам'яті не менше 512 Мбайт.
- Доступ до інтернету зі швидкістю більше ніж 1 Mbps

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.002 ТЗ | Арк. |
| | | | | | | 3 |
| Зм. | Арк. | № докум. | Підпис | Дат | | |

Пояснювальна записка
до дипломного проєкту
на тему: «Оповіщувальна комунікаційна система»

Київ - 2020 року

ЗМІСТ

| | |
|---|----|
| ВСТУП | 2 |
| РОЗДІЛ 1 | 5 |
| АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 5 |
| 1.1. Огляд оповіщувальних комунікаційних систем | 5 |
| 1.2. Вимоги до програмного забезпечення | 7 |
| Висновки до першого розділу..... | 15 |
| РОЗДІЛ 2 | 16 |
| МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 16 |
| 2.1 Моделювання..... | 16 |
| 2.2 Архітектура системи..... | 21 |
| 2.3 Конструювання системи..... | 23 |
| 2.4 Аналіз безпеки даних..... | 36 |
| Висновки до другого розділу | 37 |
| РОЗДІЛ 3 | 38 |
| АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ . | 38 |
| 3.1. Аналіз якості ПЗ..... | 38 |
| 3.2. Тестування | 38 |
| 3.3. Опис контрольного прикладу | 47 |
| Висновки до третього розділу..... | 49 |

| | | | | | | | | | |
|-----------|------|-----------------|-------|-----|--|--|-------|---------|--|
| | | | | | ІАЛЦ.467100.003 ПЗ | | | | |
| | | | | | | | | | |
| Зм. | Арк. | № докум. | Підп. | Дат | | | | | |
| Розробив | | Максимчук Д.А. | | | Оповіщувальна комунікаційна система Пояснювальна записка | Літ. | Аркуш | Аркушів | |
| Перевір. | | Виноградов Ю.М. | | | | | 1 | 57 | |
| | | | | | | НТУУ “КПІ імені Ігоря Сікорського”, ФІОТ Група ІО-64 | | | |
| Н. контр. | | Симоненко В.П. | | | | | | | |
| Затверд. | | | | | | | | | |

| | |
|---|----|
| РОЗДІЛ 4 | 50 |
| ВПРОВАДЖЕННЯ ТА СУПРОВІД ДОДАТКУ | 50 |
| 4.1. Інтеграція системи із додатком..... | 50 |
| 4.2. Розгортання додатку | 51 |
| 4.2.1. Встановлення та налаштування PostgreSQL | 51 |
| 4.2.2. Встановлення Python та його бібліотек | 52 |
| 4.2.3. Встановлення ReactJS та його компонентів | 52 |
| 4.2.4. Встановлення сучасного веб-браузера | 52 |
| 4.2.5. Налаштування та запуск додатку | 53 |
| 4.3. Розгортання додатку за допомогою Docker | 53 |
| Висновки до четвертого розділу..... | 54 |
| ВИСНОВКИ..... | 55 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 56 |
| ДОДАТКИ..... | 57 |

ВСТУП

Привітання членів організацій чи компаній з днем народження є і буде завжди актуальним. Адже це подія, яка циклічно відбувається із року в рік. Сьогодні організація міроприємств відбувається офлайн у реальному житті чи у комунікаційних системах на кшталт месенджерів. Способом нагадування є запис у блокноті, смартфоні чи онлайн календарі. Це не відмінняє той фактор, що запис може загубитися чи просто бути забутим.

Актуальність теми

Через зростання інтернет-користувачів, виникає можливість створення автоматизованої системи нагадування та організації привітання людей із днем народження.

Мета і задачі дослідження

Мета створення даної роботи – розробити зручний засіб для привітань та спростити вже існуючі варіанти, щоб зменшити вірогідність людського фактору не привітати учасника. Автоматизація процесу зменшує кількість дій, необхідні для організатора міроприємства задля запуску святкування. А система оповіщень надає змогу учасникам отримувати актуальну інформацію по привітанням колеги чи друга.

Для досягнення поставленої мети були поставлені наступні основні задачі:

- Провести аналіз та необхідність системи оповіщень;
- Створити програму відправки повідомлень із нагадуваннями;
- Розробити систему організації міроприємств;
- Розробити засоби комунікації для учасників привітань.

Практичне значення

Розроблене програмне забезпечення створюється для подальшого впровадження в організаціях чи компаніях для привітань учасників із днем народження.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 3 |

Запропонована система економить час та кількість дій організації привітань для організатора, а також слугує засобом оповіщення та зручної комунікації для учасників.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| | | | | | | 4 |
| Зм. | Арк. | № докум. | Підпис | Дат | | |

РОЗДІЛ 1

АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Огляд оповіщувальних комунікаційних систем

Оповіщувальні комунікаційні системи – це системи, створені з метою нагадування та повинні містити можливість комунікації для учасників системи.

В інтернеті можна знайти велику кількість оповіщувальних комунікаційних систем. Але здебільшого вони не містять потрібних можливостей, а саме організації привітань. Основна їх ціль полягає в оповіщенні, тобто нагадуванні про дію.

У ході дослідження, було виявлено 4 системи зі схожими за функціональністю:

- *Google Calendars;*
- *Facebook;*
- *SmartGreen;*
- *Slack.*

Слід дослідити кожен з аналогів на можливість їх застосування для організації привітань із днем народження.

Google Calendars – це найпопулярніший онлайн календар, який дає змогу створювати нагадування та відмічати події, а також додавати до них невеликий опис. Із мінусів можна відмітити те, що у сервіса відсутня можливість створити декілька нагадувань для події. Також Google Calendars має скудний функціонал для комунікації, обмежений всього лише одним описом події.

Facebook – сьогодні це найбільша соціальна мережа у світі. Вона має необхідний функціонал як для оповіщень, так і для комунікації. Перевагами є автоматичне нагадування про день народження учасника вашої сторінки, тобто друга, а також можливість створювати міроприємства із великою кількістю додаткових налаштувань таких як місце, дата, опис, зображення.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 5 |

Недоліком є неможливість отримувати повідомлення про день народження людини, яка не є учасником вашої сторінки, тобто людина яка не додана до вашого списку друзів. Це унеможлиблює процес привітання колеги чи члена іншої організації. Також основна ціль Facebook – це спілкування, а не організація міроприємств.

SmartGreen – це стартап по автоматизації привітань із днем народження. Він має можливість налаштувати автоматичне привітання, та має великий набір унікальних шаблонів листів. Але сервіс не надає учасникам платформу для комунікації.

Slack – сервіс для корпоративної комунікації між членами організації чи компанії. Більшість великих компаній, використовують його для вирішення питань. Великим плюсом є зручне створення груп та каналів комунікації, а також розвинена система оповіщень. Але Slack не містить функціоналу для відслідковування та нагадування про день народження учасників. У дипломній роботі, Slack використовується як спосіб комунікації та оповіщень учасників системи.

Таблиця 1.1 – Порівняльна характеристика проектів.

| Назва сервісу | Google Calendars | Facebook | SmartGreen | Slack |
|--|------------------|----------|------------|-------|
| Можливість створювати подію | + | + | - | + |
| Відслідковування днів народження учасників | + | + | + | - |
| Система оповіщень | - | - | - | - |

Продовження таблиці 1.1 - Порівняльна характеристика проектів.

| Назва сервісу | Google Calendars | Facebook | SmartGreen | Slack |
|--|---------------------|----------|------------|-------|
| Можливість комунікації для учасників | - | + | - | + |
| Інтеграція з базою організацій | + | - | - | + |

З таблиці 1.1, можна зробити висновок, що жоден із вище перелічених сервісів не надає повний функціонал оповіщувальної комунікаційної системи та зручного способу організації дня народження члена організації чи компанії.

1.2. Вимоги до програмного забезпечення

Варіанти використання системи наведено нижче.

Таблиця 1.2 – Варіант використання UC01

| | |
|------------|---|
| Назва | Авторизація користувача |
| Опис | Член організації чи компанії має можливість авторизуватися у додатку |
| Учасники | Користувач |
| Передумови | Інформація про користувача повинна міститись у базі організації чи компанії |
| Постумови | Користувач авторизований у системі |
| Передумови | Інформація про користувача повинна міститись у базі організації чи компанії |
| Постумови | Користувач авторизований у системі |

Продовження таблиці 1.2 – Варіант використання UC01

| | |
|---------------------|--|
| Основний сценарій | Система демонструє вікно авторизації Google Accounts Користувач вводить логін та пароль від облікового запису Google Користувач натискає кнопку “Увійти” Система авторизує користувача та перенаправляє на головну сторінку |
| Розширений сценарій | Система не авторизує користувача через неправильні дані від облікового запису Google, або при відсутності даних про користувача у базі організації чи компанії. Система демонструє повідомлення про помилку |

Таблиця 1.3 – Варіант використання UC02

| | |
|-------------------|---|
| Назва | Перегляд списку існуючих подій та списку днів народжень учасників |
| Опис | Авторизований користувач має можливість перегляду активних міроприємств та списку іменинників |
| Учасники | Користувач |
| Передумови | Користувач пройшов авторизацію у системі та перейшов на головну сторінку веб-сайту, активні події існують |
| Постумови | Список подій та іменинників |
| Основний сценарій | Система відображає список іменинників та подій. |

Таблиця 1.4 – Варіант використання UC03

| | |
|-------------------|---|
| Назва | Заповнення інформації про користувача |
| Опис | Авторизований користувач має можливість вказати інформацію про свої інтереси, бажаний подарунок, склад своєї команди, склад заступників, свої реквізити |
| Учасники | Користувач |
| Передумови | Користувач авторизований у системі, поля для введення відображено |
| Постумови | Профіль з інформацією про користувача заповнено |
| Основний сценарій | Користувач вказує інформацію про себе Система запам'ятовує та відображає дані |

Таблиця 1.5 – Варіант використання UC04

| | |
|---------------------|---|
| Назва | Створення нової події |
| Опис | Авторизований користувач може організувати привітання будь-якого учасника системи |
| Учасники | Організатор |
| Передумови | Організатор авторизувався у системі, список учасників було відображено |
| Постумови | Створено нове міроприємство, інформація про подію відображено |
| Основний сценарій | Користувач вибирає користувача, якого слід привітати із днем народження, заповнює поля про подарунок, реквізити Система створює подію та додає учасників команди авторизованого користувача до міроприємства |
| Розширений сценарій | Система не дає змогу вибрати декілька користувачів для привітання, та додати в подію іменинника |

Таблиця 1.6 – Варіант використання UC05

| | |
|-------------------|--|
| Назва | Редагування інформації про подію |
| Опис | Організатор події має можливість редагувати дані про створене міроприємство |
| Учасники | Організатор, заступник організатора |
| Передумови | Користувач авторизований у системі, подію створено, користувач є організатором, заступник організатора доданий в особистому кабінеті |
| Постумови | Інформація про подію змінена |
| Основний сценарій | Організатор переходить на сторінку редагування події, вказує нову інформацію, за потреби делегує організацію на свого заступника |

Таблиця 1.7 – Варіант використання UC06

| | |
|-------------------|---|
| Назва | Приєднання користувача системи до складу учасників події |
| Опис | Авторизований користувач може приєднатися до привітання іншого користувача |
| Учасники | Користувач |
| Передумови | Користувач авторизований у системі, подію створено, інформацію про подію відображено |
| Постумови | Користувач стає учасником міроприємства, інформацію про подію оновлено |
| Основний сценарій | Користувач натискає кнопку "підписатися на користувача" Система запам'ятовує дію та в день народження іменинника відправляє повідомлення з нагадуванням про привітання |

Продовження таблиці 1.7 – Варіант використання UC06

| | |
|---------------------|--|
| Розширений сценарій | Користувач стає учасником події за умови, що подія створена. Користувачу відправлено повідомлення з інформацією про подію та запрошення приєднатися. |
|---------------------|--|

Таблиця 1.8 – Варіант використання UC07

| | |
|---------------------|---|
| Назва | Переведення коштів учасником події |
| Опис | Учасник події має можливість вказати переведену суму на подарунок імениннику |
| Учасники | Користувач |
| Передумови | Користувач авторизований у системі Slack та є учасником вже створеної події, в інформації про міроприємство вказані реквізити організатора, повідомлення з даними про рахунок було відправлено |
| Постумови | Кошти відправлені на реквізити організатора, інформацію про подію та зібрані кошти оновлено, повідомлення про їжу відправлено, система запам'ятала користувача як учасника |
| Основний сценарій | Користувач переводить через онлайн-банк кошти на рахунок, вказаний у повідомленні про подію, загальний рахунок збільшується |
| Розширений сценарій | При неправильно вибраній сумі, яку користувач перевів по реквізитах, є можливість вказати коректну суму. Якщо користувач не перевів кошти, система на наступний день відправляє нагадування про відправку коштів |

Таблиця 1.9 – Варіант використання UC08

| | |
|---------------------|---|
| Назва | Вказівка учасника події про бажання їсти |
| Опис | Учасник події може вказати чи він буде їсти на події привітання |
| Учасники | Користувач |
| Передумови | Користувач авторизований у системі Slack та є учасником вже створеної події, кошти на подарунок було та повідомлення про їжу було відправлено |
| Постумови | Інформацію про подію та кількість людей, які будуть їсти, оновлено |
| Основний сценарій | Користувач вказує чи буде він їсти на міроприємстві |
| Розширений сценарій | Користувач проігнорував повідомлення про їжу, наступного дня система відправляє нагадування вказати статус |

Таблиця 1.10 – Варіант використання UC09

| | |
|---------------------|--|
| Назва | Комунікація учасників |
| Опис | Створюється канал у Slack з учасниками події, де вони можуть комунікувати та вирішувати питання |
| Учасники | Організатор, учасники події |
| Передумови | Організатор авторизований у системі, подію створено |
| Постумови | Канал комунікації створено, інформація про подію оновлено |
| Основний сценарій | Користувач натискає кнопку "створити канал Slack", система створює канал у Slack та додає всіх учасників події до каналу |
| Розширений сценарій | Якщо канал вже створено, є можливість видалити канал, натиснувши на кнопку "Архівувати канал" |

Таблиця 1.11 – Варіант використання UC10

| | |
|---------------------|--|
| Назва | Святкування дня народження |
| Опис | Іменинник отримує інформацію про учасників події святкування свого дня народження та сповіщає їх про відмічання свята |
| Учасники | Іменинник |
| Передумови | Користувач авторизований у системі та у Slack, подію створено, повідомлення про день народження відправлено |
| Постумови | Учасникам події відправлено повідомлення від іменинника |
| Основний сценарій | У день народження, іменинник отримує повідомлення про необхідність вказати де, як та коли буде святкувати. Перейшовши за посиланням, на головній сторінці відображено панель із інформацією про учасників та полем, де слід вказати інформацію про святкування. Після натискання кнопки "ОК", повідомлення з інформацією з поля відправляється учасникам події |
| Розширений сценарій | Подію привітання не створено, іменинник не отримує повідомлень |

Нижче наведено таблиці із функціональними вимогами до системи.

Таблиця 1.12 – Опис функціональної вимоги REQ001

| | |
|-------|---|
| Номер | REQ001 |
| Назва | Вихід із системи |
| Опис | Додаток надавати можливість виходу з особистого акаунту |

Таблиця 1.13 – Опис функціональної вимоги REQ002

| | |
|-------|--|
| Номер | REQ002 |
| Назва | Можливість вимикання повідомлень |
| Опис | У системі повинна бути функція налаштувань повідомлень, які хоче отримувати користувач |

Таблиця 1.14 – Опис функціональної вимоги REQ003

| | |
|-------|---|
| Номер | REQ003 |
| Назва | Підписування на користувачів |
| Опис | Система дозволяє підписуватись на користувачів задля подальшого інформування про дні народження |

Таблиця 1.15 – Опис функціональної вимоги REQ004

| | |
|-------|--|
| Номер | REQ004 |
| Назва | Інформація про дні народження всіх учасників системи |
| Опис | Система повинна повідомляти користувачів про всі дні народження, які відбудуться в учасників організації чи компанії |

Висновки до першого розділу

У цьому розділі було описано предметну область розробки та зроблений її детальний аналіз. Виділено вже існуючі системи, що стосуються даної предметної області. Розглянуто їх переваги, недоліки та можливості. Визначено варіанти використання та функціональні вимоги. Результати приведені у вигляді порівняльної таблиці.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| | | | | | | 15 |
| Зм. | Арк. | № докум. | Підпис | Дат | | |

РОЗДІЛ 2

МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Програмне забезпечення складається із бізнес логіки і графічного інтерфейсу та повинне забезпечувати високу доступність, гнучке масштабування. Тому доречно розділити програму на окремі модулі, які будуть відповідати за вирішення окремих логічних задач. Для забезпечення ефективної взаємодії модулів, слід використати спеціальну архітектуру *MVC* (*Model View Controller*), а також розділити систему на дві частини: клієнтську та серверну.

Шаблон *MVC* надає змогу розширювати функції програми без впливу на існуючий. А також розділяє програму на 3 частини: модель даних, відображення даних, керування даними. Відокремлення модулів дозволяє гнучко масштабувати системою.

Розподілення програми на клієнтську та серверну частини пришвидшує час завантаження сторінки та в результаті створюється 2 окремих проекта. Сервер передає дані по запиту від клієнтської частини через API (*Application Programming Interface* – програмний інтерфейс додатка), незважаючи на спосіб реалізації клієнта. Така система є взаємонезалежною та динамічною.

2.1. Моделювання

Перед етапом реалізації, потрібно виконати етап моделювання. Для основних функцій системи будуються діаграми завдяки онлайн-сервісу Draw.io. На зображеннях нижче можна побачити схеми основних процесів, при користуванні системою.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 16 |

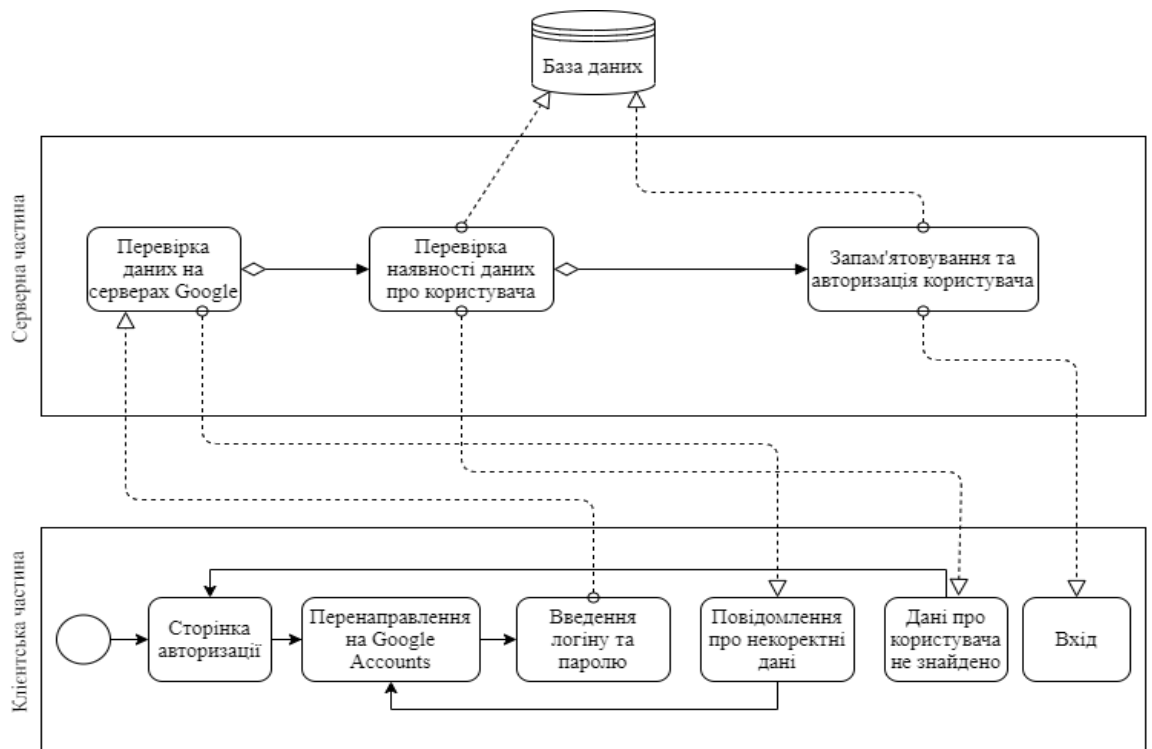


Рисунок 2.1 – Авторизація користувача системи

Послідовність етапів авторизації користувача на сайті:

- користувач системи натискає кнопку "Авторизуватись";
- введення даних Google Accounts;
- відправка даних на сервер Google;
- при правильно введених даних, відправка даних із серверу Google на сервер застосунку, інакше перенаправлення користувача на попередню сторінку;
- збереження даних на сервері застосунку, повідомлення користувача про успішну чи неуспішну авторизацію.

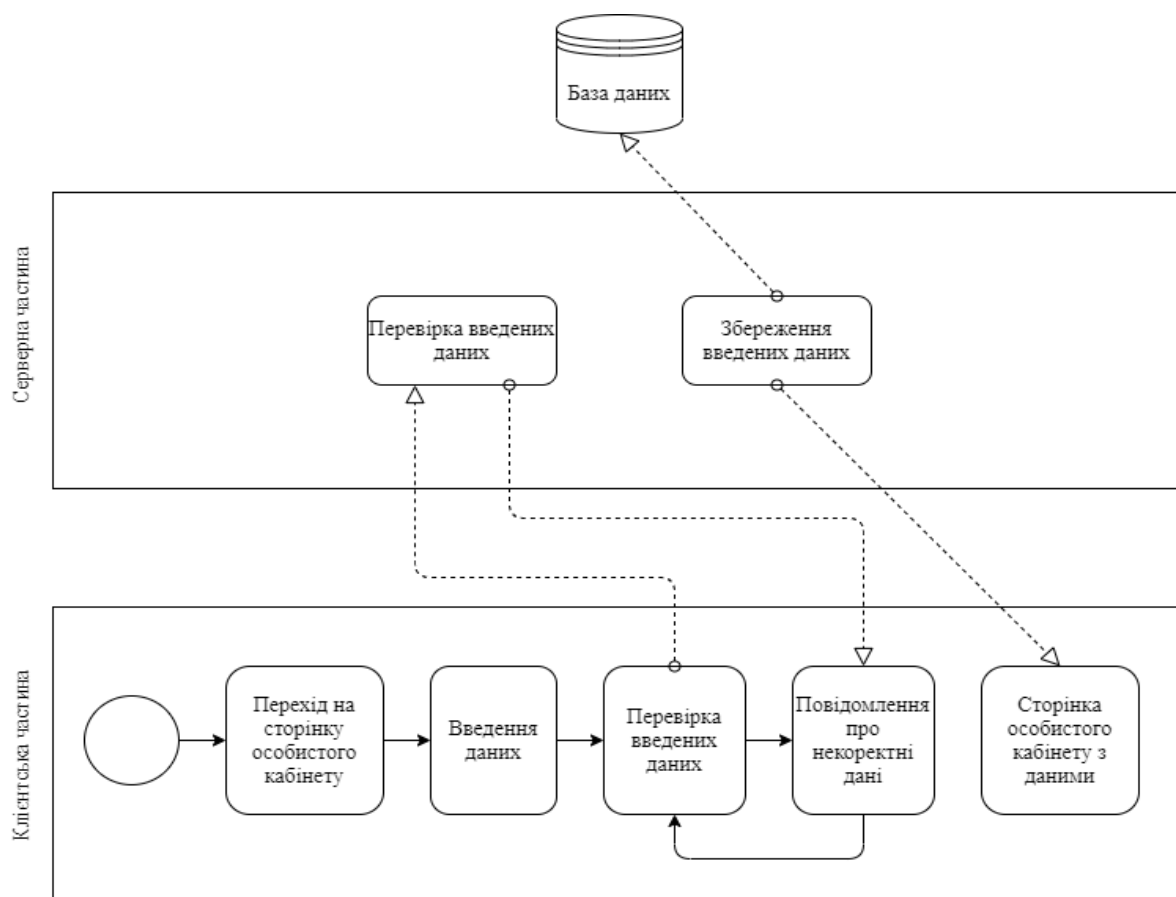


Рисунок 2.2 – Заповнення даних про користувача

Опис процесу заповнення користувачем даних про себе:

- користувач переходить на сторінку особистого кабінету;
- користувач вводить дані про себе (реквізити, інтереси, бажання, члени команди, заступники);
- перевірка даних на клієнтській частині, при успішно пройденій перевірці відправка запиту на оновлення даних на сервері;
- перевірка даних на сервері, збереження нових даних про користувача;
- користувач залишається на сторінці особистого кабінету.

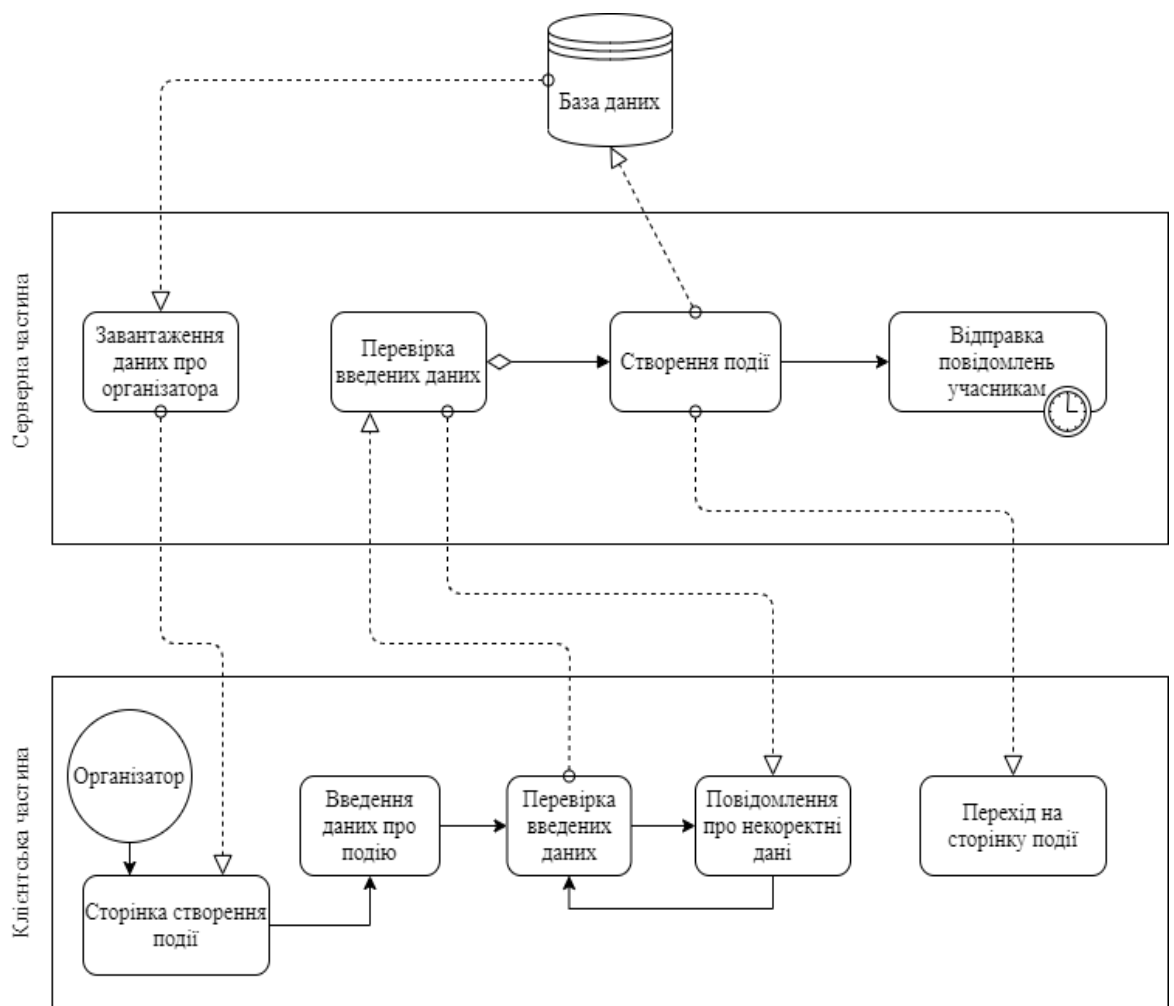


Рисунок 2.3 – Схема створення нової події

Послідовність створення міроприємства:

- організатор переходить на сторінку створення події;
- користувач вводить дані про подію;
- перевірка даних на клієнтській частині, при успішно пройденій перевірці відправка запиту на створення події на сервері;
- перевірка даних на сервері, створення події та збереження інформації про неї в базі даних;
- перенаправлення користувача на сторінку події;
- відправка учасникам команди організатора повідомлень про створену подію та запрошення приєднатися до неї.

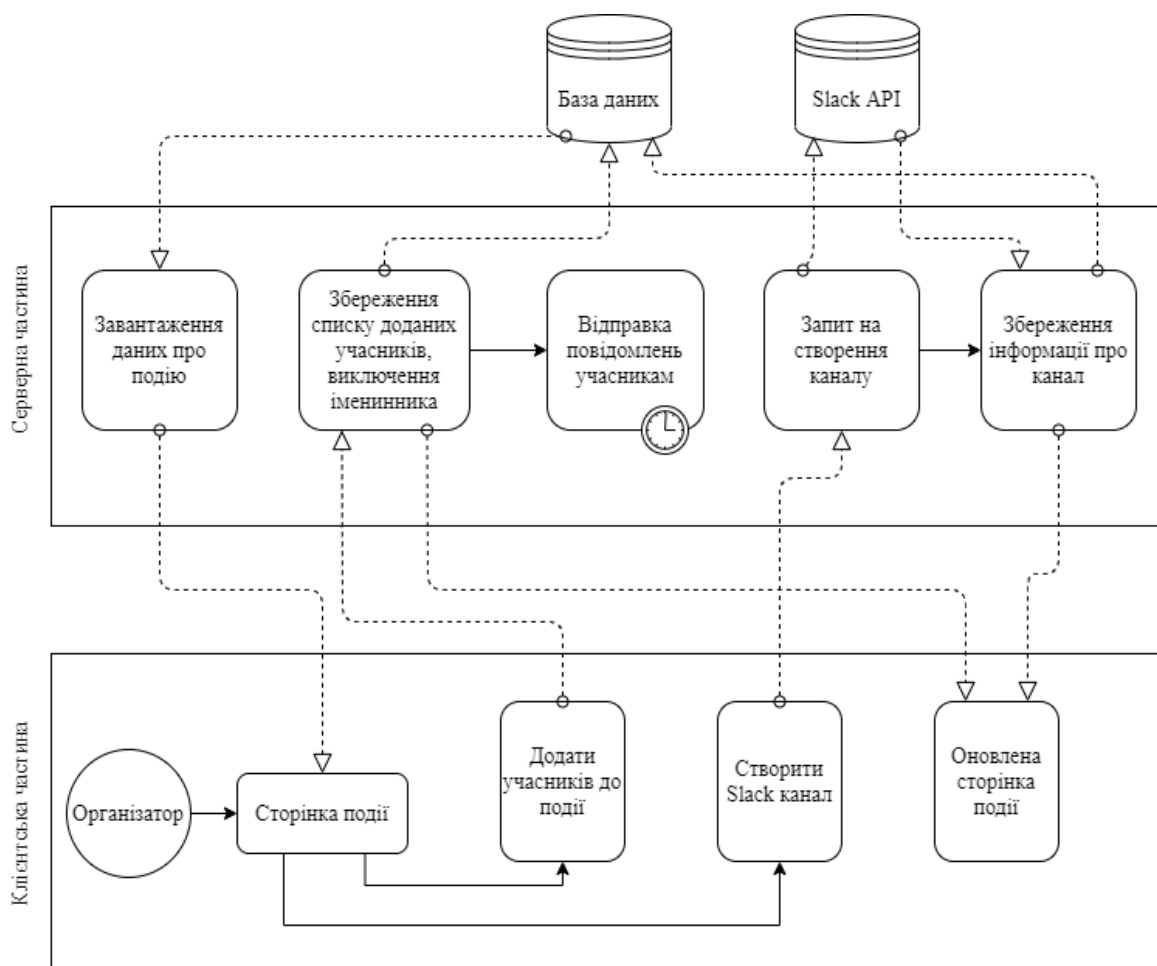


Рисунок 2.4 – Схема взаємодії з міроприємством

Опис можливих дій із подією у системі:

- *організатор переходить на сторінку події;*
- *при натисканні на кнопку "додати користувачів", має можливість вибрати нових учасників;*
- *збереження списку нових учасників, асинхронна відправка повідомлень у Slack;*
- *при натисканні на "Створити Slack канал", відправляється повідомлення із запитом на створення каналу до серверу Slack;*
- *Slack створює канал відправляє запит на сервер системи, система зберігає дані про канал та відправляє їх на клієнтську частину;*
- *користувач бачить оновлену сторінку з інформацією про подію.*

2.2. Архітектура системи

Доступ до розробленої системи є у будь-якого пристрою із доступом до інтернету та встановленим одним із браузерів останньої версії: Google Chrome, Mozilla Firefox, Safari. Зв'язок із системою відбувається за протоколу HTTPS (Hyper Text Transfer Protocol Secure) протоколу.

Для розробки клієнтської частини було використано мову JavaScript стандарту EcmaScript 6 із використанням бібліотеки з відкритим кодом React. Вибір було зроблено на користь JavaScript, тому що він не має таких потужних та динамічних аналогів. Із мінусів можна відокремити іноді непередбачуваний результат деяких операцій та неможливість типізувати змінні, адже він є мовою із слабкою типізацією, тобто змінні можуть змінювати свій тип під час виконання скрипту. Модулі проєкту було відокремлено та розділено для подальшого їх перевикористання та можливості імпортування в інші проєкти. Графічна частина містить в собі мову розмітки HTML (Hyper Text Markup Language), а також формальну мову розмітки CSS (Cascading Style Sheets) для надання сторінкам власного унікального стилю та зручності для використання користувачем. Взаємодія із серверною частиною відбувається поверх захищеного Web протоколу HTTPS протоколу та кодуванням даних у вигляді формату JSON (JavaScript Object Notation).

Серверна частина була розроблена за допомогою мови програмування Python 3 та його асинхронного фреймворку, тобто бібліотеки, з відкритим кодом AsyncIO. Переваги асинхронності дозволяють значно зменшити час виконання програм, особливо коли ПО містить інтеграції з іншими сервісами, адже час виконання запиту дорівнює часу найповільнішого запиту, замість суми часу затраченого на всі запити окремо. Для розробки було використано паттерн MVC, що економить час для розробників завдяки розділенню проєкта на декілька умовних частин. Для зберігання даних було використано об'єктно-реляційну систему керування базами даних

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 21 |

PostgreSQL. На сьогодні вона є одною із найпопулярніших СУБД (система управління базою даних) та надійних баз даних у світі.

Для швидкого та зручного розгортання системи на сервері використано GitLab CI. Це інструмент, який допомагає розгортувати або виконувати автоматично перевірку тестів, програм які слідкують за стилем коду кожного разу, коли змінюється кодова база. Також він дає змогу розгортати додаток на декілька окремих серверів одночасно, якщо працює у зв'язці із Docker та Kubernetes. Docker — інструмент для управління ізольованими Linux-контейнерами. Він надає можливість в ізольованому оточенні від інших процесів запускати систему, при цьому використовувати лише вказані у конфігурації залежності проєкта. Це гарантує однакові версії залежностей, тобто однакове виконання програми у будь-якій системі. Kubernetes — відкрита система автоматичного розгортання, масштабування та управління застосунками у контейнерах, розроблена компанією Google. У ньому описуються правила розгортання на сервері. Одного опису достатньо, щоб задати правила запуску застосунку на будь-якій кількості серверів навіть у різних датацентрах. Також Kubernetes має різні функції, які слідкують за правильним виконанням програми та сам масштабує кількість реплік програми. Тобто, якщо на сервері станеться помилка, то можна завчасно описати, що потрібно робити із цим сервером. Або якщо навантаження на сервер зросло, можна описати правила, за якими збільшити системі об'єм оперативної пам'яті чи процесорного часу. Завдяки йому можна досягти високої надійності, безперебійності, швидкодії системи.

Розроблена система інтегрована у корпоративний месенджер Slack задля комунікації із учасниками. Було створено чат-бота для відправки повідомлень користувачам, який працює як зворотній зв'язок між розроблювальним сервером та користувачем. Система відсилає запит на сервер Slack, після чого повідомлення відправляється користувачу. Схожим чином працює зв'язок між користувачем та системою: користувач системи

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 22 |

робить запит на Slack, а він в свою чергу перенаправляє цей запит на сервер системи.

Для авторизації у системі, було використано Google Accounts, адже це найпопулярніша та одна з найкращих систем, призначених для аутентифікації користувачів. Тому кожен користувач системи повинен бути зареєстрованим у Google.

2.3. Конструювання системи

Опис класів клієнтської частини наведено у таблиці 2.1. Запити із клієнтської частини описано в таблиці 2.2. Методи серверної частини наведено у таблиці 2.3. Таблиця 2.4 включає в себе методи для відправки повідомлень. В таблиці 2.5 описано таблиці бази даних.

Таблиця 2.1 – Опис класів клієнтської частини

| Назва класу | Опис |
|-------------|--|
| App | Головний клас, який є точкою входу в додаток. У ньому об'явлені всі компоненти клієнтської частини |
| Config | Клас конфігурування додатку, містить константи, які ініціалізуються при запуску та регулюються в залежності від середовища |
| Routes | Опис констант з URL шляхами додатку |
| UserList | Клас, який відповідає за бізнес-логіку відображення списку учасників системи |
| UserCard | Містить в собі логіку відображення даних користувача у вигляді картки |
| Profile | Компонент, відповідальний за відображення сторінки особистого кабінету користувача. Містить бізнес-логіку оновлення даних користувача системи та компоненти Preferences та DebitCartList |

Продовження таблиці 2.1 – Опис класів клієнської частини

| Назва класу | Опис |
|---------------|--|
| Preferences | Підкомпонент Profile, містить логіку обробки інтересів та бажань користувача |
| DebitCartList | Підкомпонент, відповідальний за обробку даних про реквізити, тобто банківських карт |
| Users | Клас, який являє собою модальне вікно у браузері, оброблює логіку вибору користувачів зі списку UserList |
| SetPlace | Компонент, який є модальним вікном, містить бізнес-логіку встановлення місця та дати проведення події іменинником |
| Logout | Клас із логікою виходу користувачів із системи |
| Login | Компонент авторизації користувачів. Слугує проміжною частиною між сервером системи та сервером Google Accounts |
| Header | Клас відповідальний за відображення верхньої частини сторінок у браузері |
| Footer | Компонент відповідальний за відображення нижньої частини сторінок у браузері |
| Error | Клас обробки помилок. Містить в собі список кодів та повідомлень про помилки |
| Event | Компонент сторінки події. Відображає інформацію про міроприємство. Має функціонал відображення карток з інформацією про користувачів, можливість додавання нових членів. |

Продовження таблиці 2.1 – Опис класів клієнської частини

| Назва класу | Опис |
|---------------------|--|
| Checklist | Компонент створення події. Відображає карти з інформацією про користувачів, вибір іменинника, поля вводу даних про міроприємство |
| ExistEventChecklist | Клас, який містить бізнес-логіку редагування інформації про вже існуючу подію |
| Dashboard | Компонент головної сторінки веб-сайту. Відображає інформацію про сайт, карточки користувачів з інформацією про них |
| BirthdayTodayWindow | Підклас Dashboard із логікою привітання користувача, у якого день народження. В собі викликає компонент SetPlace |

Далі наведено опис списку запитів із клієнської частини на серверну. Запити відбуваються асинхронно за допомогою REST (*Representational State Transfer*) поверх HTTPS протоколу.

Таблиця 2.2 – Опис запитів із клієнської частини

| Назва запиту | Опис |
|--------------------------|---|
| checklistData | Запит на завантаження даних для відображення сторінки створення події. Повертає дані про іменинника та організатора |
| existEventChecklistData | Завантаження даних про вже створену подію задля подальших маніпуляцій з ними, у тому числі редагування |
| changeChecklistOrganizer | Запит на сервер з метою зміни організатора події |

Продовження таблиці 2.2 – Опис запитів із клієнської частини

| Назва запиту | Опис |
|---------------------|--|
| getDepartmentsUsers | Метод, який завантажує дані про групи або департаменти всередині організації |
| getUsersProfileList | Запит на список даних із особистого кабінету користувачів |
| usersGetOrUpdate | Завантаження або зміна даних про користувачів |
| getUserProfile | Запит на отримання даних особистого кабінету користувача |
| updateUserProfile | Запит на зміну даних користувача в особистому кабінеті |
| getCards | Завантаження даних про кредитні картки користувача |
| addCards | Запит на створення нової кредитної картки |
| deleteCard | Запит на видалення кредитної картки користувача |
| logout | Запит на сервер системи, щоб він видалив дані про сесію користувача та |
| setLoggedStatus | Метод, який робить запит на сервер і встановлює статус авторизації |
| authorizeUser | Запит на авторизацію користувача та збереження його даних про сесію |
| setEventPlace | Завантаження даних про відмічання події на сервер |
| getEatersNumber | Запит на підрахунок кількості людей, які будуть їсти на події |
| getEventData | Отримання всіх даних про конкретну подію |

Продовження таблиці 2.2 – Опис запитів із клієнської частини

| Назва запиту | Опис |
|-------------------|---|
| createGroup | Запит на створення нового каналу в Slack |
| connectToGroup | Приєднання користувача до існуючої групи в Slack |
| archiveGroup | Запит на архівування групи Slack |
| userSubscriptions | Отримання даних про підписки користувача на інших учасників системи |
| getUsersBirthdays | Запит на список учасників, у яких скоро день народження |
| getUsersEvents | Завантаження списку активних подій у системі |
| deleteEvent | Запит видалення події |
| joinToEvent | Запит на приєднання учасника до події |

Опис методів серверної частини. Кожний метод є контролером, аргументом якого є запит із даними отриманий від клієнтської частини.

Таблиця 2.3 – Опис методів серверної частини

| Назва методу | Опис |
|-------------------------|--|
| google_oauth_callback | Метод обробки запиту, отриманого від Google Accounts про успішну чи не успішну авторизацію |
| logout | Контролер, який містить в собі логіку видалення даних про сесію користувача |
| dashboard | Бізнес-логіка підготовки та обробки даних, які потім будуть відображені на головній сторінці |
| subscriptions | Метод завантаження даних із бази про підписки користувачів |
| active_organized_events | Повертає список активних подій у системі |

Продовження таблиці 2.3 – Опис методів серверної частини

| Назва методу | Опис |
|-------------------------------|--|
| delete_event | Контролер видалення події з системи |
| join_to_event | Метод по запиту додає користувача до події |
| get_event_memers | Обробка запиту на отримання користувачів, які належать конкретній події |
| add_event_members | Контролер створення запису про нового користувача події |
| get_event_data | Метод повертає дані про подію, які потім відображаються на сторінці міроприємства |
| set_birthday_place_and_date | Метод збереження інформації про місце та час проведення події |
| get_birthday_number_eaters | Обробка запиту на отримання кількості людей, які будуть їсти на події |
| create_group | Слугує проміжною частиною, отримує запит на створення нової групи у Slack та відправляє запит на створення групи на сервер Slack |
| connect_to_group | Метод додавання користувача до групи Slack |
| archive_group | Обробка запиту на архівування Slack групи |
| create_or_update_organization | Метод створює перевіряє отримані в запиті дані та створює подію або оновлює дані про неї |
| get_organization | Повертає дані, які необхідні для створення нової події. Тобто інтереси іменинника та дані про організатора |
| change_organizer | Метод обробки запиту на зміну організатора події |
| profile | Метод повертає дані для особистого кабінету користувача |

Продовження таблиці 2.3 – Опис методів серверної частини

| Назва методу | Опис |
|----------------------------|--|
| get_users_profiles_by_list | Повертає дані з особистих кабінетів списку користувачів |
| get_event_data | Метод повертає дані про подію, які потім відображаються на сторінці міроприємства |
| carts | Обробка запиту отримання даних про кредитні картки користувача |
| get_team | Метод дістає із бази інформацію про команду користувача |
| add_team | Додавання користувача до команди учасників привітання |
| remove_team | Метод видалення користувача з команди учасників |
| add_alternates | Додавання користувача до списку заступників події |
| remove_alternates | Видалення користувача із списку заступників |
| get_organization | Повертає дані, які необхідні для створення нової події. Тобто інтереси іменинника та дані про організатора |
| change_organizer | Метод обробки запиту на зміну організатора події |
| create_event_member | Створення нового учасника події |
| create_subscription | Метод створення нового запису про підписку користувача на учасника системи |
| delete_subscription | Видалення запису про підписку |
| insert_money | Метод обробки запиту від Slack на відправлення коштів на подарунок імениннику учасником |

Продовження таблиці 2.3 – Опис методів серверної частини

| Назва методу | Опис |
|-----------------------|--|
| cancel_money_callback | Обробка Slack запиту на зміну відправленої суми користувачем |
| update_eat_status | Метод обробляє запит на зміну статусу чи буде користувач їсти на міроприємстві |

Опис методів для відправки повідомлень із серверної частини до Slack.

Таблиця 2.4 – Опис методів відправки повідомлень

| Назва методу | Опис |
|--------------------------------|---|
| heads_command_empty_notify | Відправка повідомлення користувачу, що його команда не заповнена в особистому кабінеті |
| member_accept_invite_notify | Відправка повідомлення організатору з інформацією, що користувач приєднався до події |
| gift_raised_notify | Метод відправлення повідомлення з інформацією про те, що кошти на подарунок зібрані |
| birthday_fill_profile_notify | Відправлення користувачу системи повідомлення із необхідністю заповнити профіль, тобто реквізити, інтереси та бажання |
| birthday_date_and_place_notify | Метод відправки повідомлення імениннику із закликом заповнити інформацію про дату та час святкування |
| event_organizer_notify | Повідомлення користувачу із пропозицією стати організатором дня народження учасника своєї команди |

Продовження таблиці 2.4 – Опис методів відправки повідомлень

| Назва методу | Опис |
|-------------------------------|--|
| event_change_organizer_notify | Відправка повідомлення про зміну організатора події |
| event_link_notify | Відправка повідомлення із інформацією про посилання на подію на сайті |
| event_send_money_notify | Повідомлення учаснику події із закликом відправити кошти на подарунок імениннику |
| send_team_msg | Відправка повідомлень учасникам команди про день народження одного із учасників |
| send_subscribers_birthday_msg | Повідомлення для підписників про день народження людини, на яку вони підписані |
| send_daily_msg | Відправка щоденних повідомлень із списком днів народжень учасників системи |

Опис таблиць бази даних.

Таблиця 2.5 – Опис моделі бази даних

| Таблиця | Запис | Тип | Опис |
|---------|---------------------|---------|-------------------------------|
| Users | id (первинний ключ) | Integer | Унікальний ідентифікатор |
| Users | vulyk_id | Integer | ID у базі системи організації |
| Users | first_name | String | Ім'я користувача |
| Users | last_name | String | Фамілія |
| Users | birthday | Date | Дата народження |
| Users | email | String | Емейл користувача |
| Users | image | String | Посилання на фото користувача |
| Users | birthday | Date | Дата народження |

Продовження таблиці 2.5 – Опис моделі бази даних

| Таблиця | Запис | Тип | Опис |
|---------|----------------|---------|---|
| Users | email | String | Емейл користувача |
| Users | image | String | Посилання на фото користувача |
| Users | image_hash | String | Хеш-код фотографії користувача |
| Users | project_id | Integer | Номер департаменту в організації |
| Users | slack_id | String | ID діалогу з користувачем у сервісі Slack |
| Users | wishes | String | Бажання користувача |
| Users | interests | String | Інтереси |
| Users | prefer_money | Boolean | Флаг чи хотів би користувач отримати конверт |
| Users | dont_celebrate | Boolean | Флаг чи буде відмічати користувач день народження |
| Users | dont_eat_meat | Boolean | Флаг чи їсть користувач м'ясо |
| Users | dr_today | Boolean | Флаг чи буде користувач отримувати повідомлення у кого сьогодні день народження |

Продовження таблиці 2.5 – Опис моделі бази даних

| Таблиця | Запис | Тип | Опис |
|---------------|---------------------|---------|---|
| Users | dr_week | Boolean | Флаг чи буде користувач отримувати повідомлення у кого день народження протягом тижня |
| Users | project_name | String | Назва департаменту користувача |
| Users | slack_user_id | String | ID користувача у сервісі Slack |
| Commands | id (первинний ключ) | Integer | Унікальний ідентифікатор |
| Commands | head_id | Integer | ID користувача, який є засновником команди |
| Commands | member_id | Integer | ID користувача, який є учасником команди |
| Subscriptions | id (первинний ключ) | Integer | Унікальний ідентифікатор |
| Subscriptions | follower_id | Integer | ID користувача, який підписався |
| Subscriptions | following_id | Integer | ID користувача, на якого підписались |
| Alternates | id (первинний ключ) | Integer | Унікальний ідентифікатор |
| Alternates | organizer_id | Integer | ID користувача, який додав заступника |

Продовження таблиці 2.5 – Опис моделі бази даних

| Таблиця | Запис | Тип | Опис |
|---------------|---------------------|---------|--|
| Alternates | alternative_id | Integer | ID користувача, якого вказали заступником |
| Debit_cards | id (первинний ключ) | Integer | Унікальний ідентифікатор |
| Debit_cards | user_id | Integer | ID користувача, який додав кредитну карту |
| Debit_cards | bank_name | String | Назва банку, який обслуговує карту |
| Debit_cards | number | String | Номер карти |
| Events | id (первинний ключ) | Integer | Унікальний ідентифікатор |
| Events | user_id | Integer | ID користувача, якого вказали іменинником |
| Events | organizer_id | Integer | ID користувача, який є організатором події |
| Events | gift | String | Назва подарунка |
| Events | price | Integer | Ціна подарунка |
| Events | slack | String | ID групи у системі Slack |
| Events | is_active | Boolean | Флаг чи активна подія |
| Event_members | id (первинний ключ) | Integer | Унікальний ідентифікатор |
| Event_members | user_id | Integer | ID користувача, що є учасником події |
| Event_members | event_id | Integer | ID створеної події у системі |

Продовження таблиці 2.5 – Опис моделі бази даних

| Таблиця | Запис | Тип | Опис |
|---------------|----------------------|---------|--|
| Event_members | event_id | Integer | ID події |
| Event_members | present | Boolean | Флаг чи буде користувач присутнім на міроприємстві |
| Event_members | will_eat | Integer | Статус чи буде їсти користувач |
| Moneys | id (первинний ключ) | Integer | Унікальний ідентифікатор |
| Moneys | event_id | Integer | ID події, в якій було відправлено кошти |
| Moneys | event_member_user_id | Integer | ID учасника події, який відправив кошти |
| Moneys | amount | Integer | Відправлена сума |
| Departments | id (первинний ключ) | Integer | Унікальний ідентифікатор |
| Departments | vulyk_dep_id | Integer | ID у системі організації чи компанії |
| Departments | name | String | Назва департаменту у системі організації |
| Departments | head_id | Integer | ID користувача, який є головним у департаменті |
| Departments | parent_id | Integer | ID департаменту, який є департаментом-батьком |

Продовження таблиці 2.5 – Опис моделі бази даних

| Таблиця | Запис | Тип | Опис |
|-------------|---------------------|---------|--------------------------------|
| Event_cards | id (первинний ключ) | Integer | Унікальний ідентифікатор |
| Event_cards | event_id | Integer | ID події, до якої додано карту |
| Event_cards | card_id | Integer | ID карти, яку було додано |

2.4. Аналіз безпеки даних

Завдяки сервісу Google Accounts, паролі користувачів зберігаються на серверах Google, тому зловмиснику аби отримати доступ до облікового запису, потрібно зламати систему Google. Авторизація відбувається на основі відповіді від Google на запит "авторизуватися", якщо вона позитивна, то система запам'ятовує користувача, записуючи до Cookies (частина даних, які передаються у HTTP запиті) унікальне значення, яке передається у тілі кожного запиту.

Доступ до бази даних PostgreSQL відбувається за допомогою імені бази, логіну та паролю, при цьому, до неї можуть під'єднуватися тільки системи із кластеру, які мають на це дозвіл. Це унеможливорює перебір паролю до бази.

Зв'язок із базою організації чи компанії задля отримання даних про членів, відбувається за допомогою API інтерфейса, доступ до якого можна отримати лише за наявності токена, який виділяється кожній системі-користувачу. Цей токен зберігається на сервері закодований за допомогою base64. Так само працює зв'язок із Slack.

Висновки до другого розділу

У цьому розділі було обрано інструменти для розробки та проведено їх аналіз. Описано використані технології та застосовані фреймворки, а також методи архітектури додатку, які спрощують та пришвидшують реалізацію програми. На етапі моделювання, було розроблено схеми взаємодії основних процесів взаємодії користувача із системою привітань із днем народження. Було розділено розробку програми на дві частини: клієнтську та серверну. Описано способи їх взаємодії, класи та методи. Було створено таблиці бази даних, які використовує додаток, а також проведено аналіз безпеки даних.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| | | | | | | 37 |
| Зм. | Арк. | № докум. | Підпис | Дат | | |

РОЗДІЛ 3

АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Тестування є важливим етапом в процесі розробки програмного забезпечення у зв'язку із специфікою роботи системи, а саме: постійним обміном даними між клієнтською та серверною частинами, а також через взаємодію додатка із іншими системами завдяки API інтерфейсу. Система повинна гарантувати безперебійний та стабільний обмін даними в системі та користувачом.

Тестування програмного забезпечення — це проведення контролю над якістю продукту, перевірка відповідності між фактичною та запланованою поведінкою програми з використанням кінцевого набору тестів.

Тестування не може повністю забезпечити правильність роботи програмного забезпечення та виявити помилки у всіх можливих випадках. Тестування може лише порівняти поведінку програми у різних ситуаціях.

3.2 Тестування

Для тестування було використано пристрій із встановленою операційною системою Windows 8, під'єднаним до мережі інтернет та встановлено сучасний браузер Chrome версії 83.0.4103. Систему заделегідь було розгорнуто у кластері за допомогою описаних технологій Kubernetes та Docker на сервері з операційною системою Ubuntu 16.04 заявки CI системи GitLab.

Для контролю правильності роботи системи у процесі тестування розроблено загальні та функціональні тест-кейси для серверної та клієнтської частин додатка. Приклади тестів наведено у таблиці 3.1 та 3.2. У таблиці 3.3 можна побачити звіт по тестуванню. Також буде проведено контрольний тест-приклад.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 38 |

Таблиця 3.1 – Загальні тест-кейси

| Назва тесту | Опис |
|-------------|--|
| GTC-1 | Перевірити роботу авторизації у системі, натиснувши на кнопку "Авторизуватися" |
| GTC-2 | Перевірити коректність відображення даних про користувача, натиснувши на ім'я користувача у верхній лівій частині веб-сторінки |
| GTC-3 | Перевірка коректності зміни налаштувань профілю на сторінці особистого кабінету, змінивши дані у полях інтересів, побажань, реквізитів |
| GTC-4 | Перевірити роботу зміни членів команди та заступників, додавши нових членів та видаливши старих |
| GTC-5 | Перевірити коректність роботи кнопки "Вихід" у верхній лівій частині сторінки |
| GTC-6 | Перевірити роботу сторінки створення події, натиснувши на кнопку "Організувати привітання" |
| GTC-7 | Перевірити коректність роботи полів введення інформації про подію |
| GTC-8 | Перевірити можливість додавання учасників по створеній події, натиснувши на кнопку "Додати учасників" |
| GTC-9 | Перевірити коректність роботи створення каналу комунікації для події, натиснувши на кнопку "Створити Slack канал" на сторінці події |
| GTC-10 | Перевірити коректність роботи видалення каналу комунікації для події, натиснувши на кнопку "Архівувати Slack канал" на сторінці події |
| GTC-11 | Перевірити роботу полів та кнопок іменинника, на головній сторінці |

Таблиця 3.2 – Функціональні тест-кейси

| Назва тесту | Опис |
|-------------|---|
| FTC-1 | Перевірити коректність вивантаження даних із бази організації чи компанії та правильне їхнє збереження у системі |
| FTC-2 | Перевірити коректність валідації поля із інформацією про реквізити, а саме номер банківської карти. Інформація не зберігається, якщо номер карти не є 16 значним числом |
| FTC-3 | Перевірити роботу функціоналу вибору користувачів із модального вікна зі списком учасників організації чи компанії |
| FTC-4 | Перевірити правильність формування списків учасників у яких день народження "сьогодні", "на цьому тижні", "на наступному тижні" |
| FTC-5 | Перевірити коректність визначення сьогоднішніх іменинників та роботу вікна із закликом вказати інформацію про святкування дня народження |
| FTC-6 | Перевірити коректність відправки повідомлень учасникам події у Slack: приєднатися до події, перевести кошти, вказати статус про їжу |
| FTC-7 | Перевірити коректність відправки повідомлень організатору події у Slack: користувач приєднався до події, кошти на подарунок зібрано |
| FTC-8 | Перевірити коректність відправки повідомлень імениннику у Slack: нагадування вказати час та місце проведення святкування |
| FTC-9 | Перевірити роботу автоматичного створення події святкування дня народження |

Звіт тестування загальних тест-кейсів наведено у таблиці 3.3.

Таблиця 3.3 – Звіт тестування загальних тестів

| Назва тесту | Очікуваний результат | Фактичний результат | Статус |
|-------------|--|--|----------|
| GTC-1 | Після переходу на сторінку Google Accounts та введення пошти та паролю, відбувається авторизація у системі | Після введення логіну та паролю, користувача авторизовано та з'являється доступ до особистого профілю | Пройдено |
| GTC-2 | Після натискання на кнопку з ім'ям користувача, відбувається перехід на сторінку із даними користувача | Відбувся перехід на сторінку користувача, де видно дані авторизованого користувача | Пройдено |
| GTC-3 | Після заповнення полів із інтересами та побажаннями, відбувається збереження даних у системі, дані на сторінці профілю оновлюються | Після заповнення полів із даними відбулось збереження до бази даних. Оновивши сторінку, дані не зникли | Пройдено |
| GTC-4 | Додавши користувачів до списку учасників команди або заступників, відбудеться збереження до бази даних та данні та сторінці оновляться | Після додавання користувачів та оновлення сторінки, дані про учасників збереглись | Пройдено |

Продовження таблиці 3.3 – Звіт тестування загальних тестів

| Назва тесту | Очікуваний результат | Фактичний результат | Статус |
|-------------|---|--|----------|
| GTC-5 | Після натискання "Вийти", система забуває користувача та користувач не матиме доступу до особистого кабінету | Після натискання на кнопку "Вийти", система забула користувача та користувач не має доступу до особистого кабінету | Пройдено |
| GTC-6 | Після натискання на кнопку створення події, відбувається перенаправлення на сторінку створення події | Після натискання на кнопку створити подію, користувача було перенаправлено на сторінку створення події | Пройдено |
| GTC-7 | У полі реквізитів повинні бути реквізити авторизованого користувача, у поле "Сума" можна вводити тільки числа | У полі з реквізитами відображено номер карти авторизованого користувача, у поле "Сума" можна ввести тільки число | Пройдено |
| GTC-8 | Після натискання на кнопку з ім'ям користувача, відбувається перехід на сторінку із даними користувача | Відбувся перехід на сторінку користувача, де видно дані авторизованого користувача | Пройдено |

Продовження таблиці 3.3 – Звіт тестування загальних тестів

| Назва тесту | Очікуваний результат | Фактичний результат | Статус |
|-------------|--|--|----------|
| GTC-9 | Після натискання на кнопку "Створити Slack канал" в системі Slack буде створено канал | Після натискання на "Створити Slack канал" було створено канал у Slack | Пройдено |
| GTC-10 | Після натискання на кнопку "Архівувати Slack канал", канал зникає із системи Slack та кнопка повертається у початкове значення | Після натискання на кнопку "Архівувати Slack канал", канал зник із системи Slack, на кнопці змінився текст на "Створити Slack канал" | Пройдено |
| GTC-11 | Після введення даних про святкування, учасникам прийде повідомлення із введеними даними | Після заповнення полів, до Slack прийшло повідомлення із введеною інформацією | Пройдено |

Опис звіту по тестуванню функціональних тестів описано нижче.

Тест FTC-1:

Очікуваний результат: після видалення користувача із бази організації чи компанії, користувач видаляється із системи.

Фактичний результат: після видалення користувача із системи організації, користувача було видалено з бази системи.

Тест FTC-2:

Очікуваний результат: дозволяється ведення лише 16 цифр.

Фактичний результат: система не дає змогу додати неправильний номер банківської карти, було додано карту із 16 значним номером.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 43 |

Тест FTC-3:

Очікуваний результат: на сторінці створення події дозволяється вибір лише одного користувача, тобто іменинника, на всіх інших – будь-яку кількість.

Фактичний результат: на сторінці створення події, якщо вибрати декілька користувачів, то система відобразить відповідну помилку. На всіх інших сторінках, можна вибрати декілька учасників. На рисунку 3.1 показана помилка, яку було відображено користувачу.

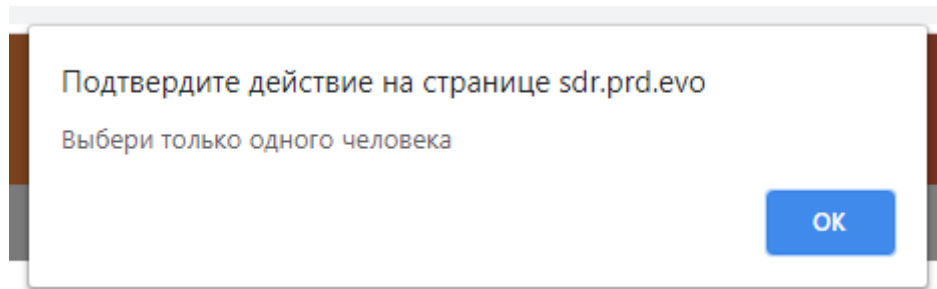


Рисунок 3.1 – Повідомлення про помилку

Тест FTC-4:

Очікуваний результат: на головній сторінці у блоці "Сьогодні" відображено користувачів, у яких сьогодні день народження. У блоці "На цьому тижні" користувачі, у яких дата народження буде протягом тижня. У блоці "На наступному тижні" відображено користувачів, день народження яких протягом наступного тижня.

Фактичний результат: на головній сторінці, у блоці "Сьогодні" відображені користувачі, у яких дата дня народження співпадає з поточною датою, у "На цьому тижні" користувачі, день народження яких відбудеться протягом тижня, у блоці "На наступному тижні" відображені користувачі, день народження яких відбудеться протягом наступного тижня.

Тест FTC-5:

Очікуваний результат: у авторизованого користувача, якщо у нього сьогодні день народження, на головній сторінці видно блок із полями, де можна вказати інформацію про святкування.

Фактичний результат: на головній сторінці відображено блок із полями, де можна вказати інформацію про святкування. При натисканні на кнопку

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 44 |

"Куди приходити", з'являється поле, куди можна ввести інформацію про подію. Повідомлення із вказаними даними, приходять учасникам, які приєдналися до привітання. На рисунку 3.2 можна побачити блок іменинника.



Привет, Сокровище! От всей души поздравляем тебя с Днем Рождения! Кликай Куда подходит, чтобы коллеги узнали где и когда тебя можно поздравить.

Будут кушать: 0
Не кушают мясо: 0

Куда приходит

Рисунок 3.2 – Блок із можливістю вказати місце та час святкування

Тест FTC-6:

Очікуваний результат: при додаванні нового користувача до події, учасник отримує відповідне повідомлення із можливістю приєднатися до події. Приєднавшись, в нього є змога вказати налаштування учасника – чи буде він їсти та вказати суму, яку він перевів.

Фактичний результат: додавши учасника системи до події, у Slack прийшло повідомлення із можливістю приєднатися. Приєднавшись, було отримано повідомлення про суму на подарунок, вказавши суму, прийшло повідомлення про їжу.

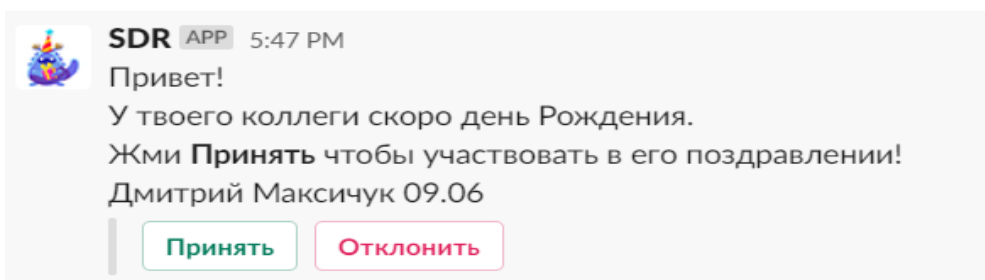


Рисунок 3.5 – Slack повідомлення із можливістю приєднатися до події

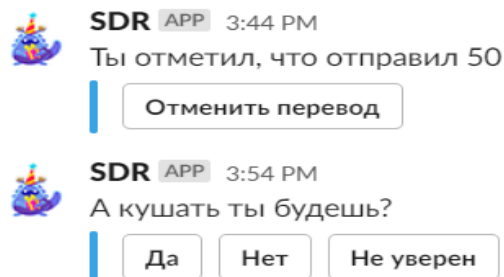


Рисунок 3.6 – Slack повідомлення учаснику події зі статусом про їжу

Тест FTC-7:

Очікуваний результат: організатор отримує повідомлення, що в учасника його команди скоро день народження, також інформацію про учасників, які долучилися до події.

Фактичний результат: організатор отримав нагадування про день народження учасника своєї команди, а також про учасників, які приєдналися. На рисунках 3.7 та 3.8 можна побачити повідомлення, які отримав організатор.

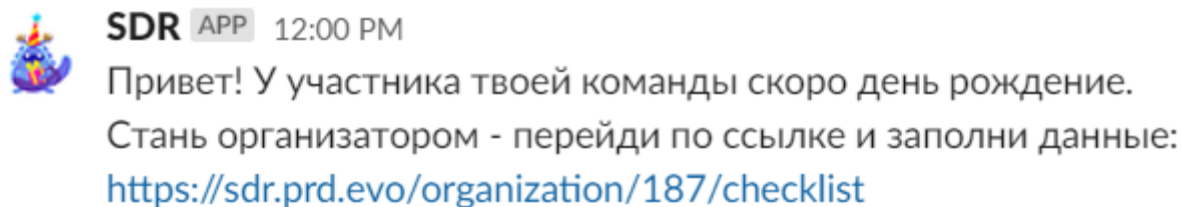


Рисунок 3.7 – Slack нагадування про день народження учасника



Рисунок 3.8 – Slack повідомлення про нового учасника

Тест FTC-8:

Очікуваний результат: користувач системи, у день свого дня народження отримує повідомлення в Slack вказати дату і місце святкування на сайті.

Фактичний результат: іменинник отримав повідомлення із закликом вказати дату та місце святкування дня народження.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 46 |



SDR APP 10:00 AM

Привет! У тебя сегодня День Рождения.

Не забудь указать коллегам время и место проведения мероприятия.

Рисунок 3.9 – Нагадування імениннику в Slack

Тест FTC-9:

Очікуваний результат: для користувачів, у яких день народження через 14 днів, автоматично створилася подія. Організатору події відправляється повідомлення у Slack із посиланням на подію на сайті.

Фактичний результат: подія була створена автоматично о третій годині ночі, організатору було відправлено повідомлення із посиланням, як показано на рисунку 3.7. При переході за посиланням, користувач редагує створену системою подію, вказуючи дані для організації події: подарунок, реквізити, суму.

3.3 Опис контрольного прикладу

Розглянемо детальніше тест кейс GTC-4 з таблиці 3.1: перевірити роботу зміни членів команди та заступників, додавши нових членів та видаливши старих. Контрольний приклад описано у таблиці 3.4.

Таблиця 3.4 – Опис контрольного прикладу

| | |
|--|--|
| Мета тесту | Перевірка роботи зміни членів команди та заступників |
| Початковий стан | Відкрита сторінка особистого кабінету, користувач авторизований у системі |
| Вхідні дані | Користувачі системи, які будуть добавлені: Дмитро Максимчук, Олексій Горбов |
| Схема проведення тесту | Натиснути "додати користувачів" на панелі членів команди та заступників |
| Очікуваний результат | У блоках "члени команди" та "заступники" буде відображено користувача системи Олексій Горбов |
| Стан програмного продукту після проведення випробувань | У блоках "члени команди" та "заступники" коректно відображено користувача системи Олексій Горбов |

На рисунку 3.10 зображено результат виконання контрольного прикладу. Учасником команди та заступником став користувач, якого було обрано у модальному вікні вибору.

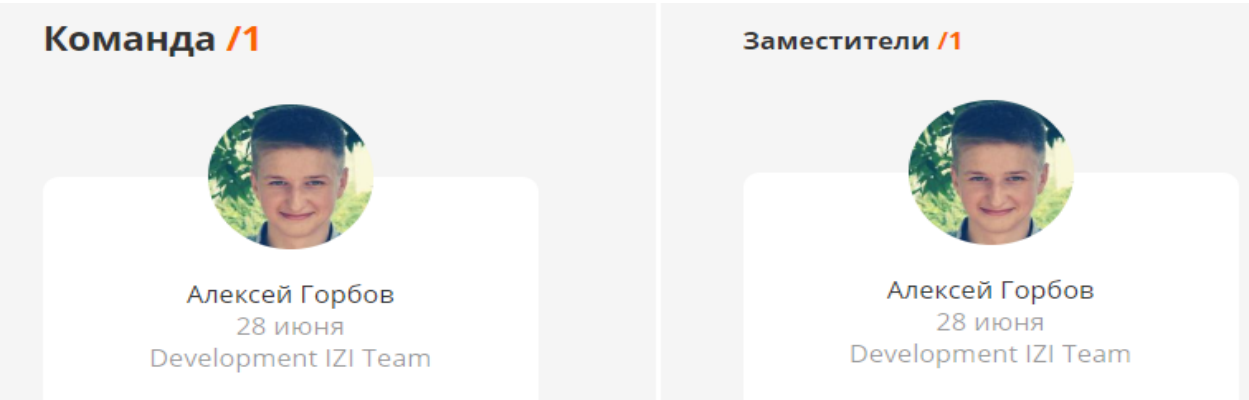


Рисунок 3.10 – Результат контрольного прикладу

Висновки до третього розділу

У цьому розділі було проведено аналіз якості програмного забезпечення, описані загальні та функціональні тест кейси для варіантів використання системи. Було зроблено тестування системи, результати наведені у пунктах 3.2 та 3.3. З'ясовано, що очікуваний і фактичний результат роботи співпадає, а якість програмного продукту задовільна.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| | | | | | | 49 |
| Зм. | Арк. | № докум. | Підпис | Дат | | |

РОЗДІЛ 4

ВПРОВАДЖЕННЯ ТА СУПРОВІД ДОДАТКУ

4.1 Інтеграція системи із додатком

Розроблювальний додаток є простим в інтеграції із вже існуючими системами компаній чи організацій. Щоб налагодити зв'язок між програмою для привітань із днем народження та системою, необхідно створити API для доступу до даних про користувачів системи. Додаток повинен знати про користувачів таку інформацію:

- *Ім'я та фамілія користувача;*
- *Дата народження;*
- *ID користувача у системі організації чи компанії;*
- *Електронну адресу користувача;*
- *Посилання на зображення користувача;*
- *ID у системі Slack;*
- *ID підрозділу у системі організації чи компанії.*

Також додатку необхідні дані про підрозділи організації, а саме:

- *ID у системі організації чи компанії;*
- *Назва департаменту;*
- *ID користувача, який є головним в департаменту;*
- *ID дочірнього департаменту, якщо такий є.*

Додаток кожного дня о третій годині ночі звертається через API до системи компанії чи організації та отримує дані про користувачів і департаменти у JSON форматі, а далі записує їх до своєї бази даних. Дані, необхідні для підключення до API, вказуються у файлі конфігурації додатку – *config.yaml* в секції *vulyk*.

Приклад налаштувань API:

```
vulyk_department_url: { посилання на API з даними про департаменти }  
vulyk_user_info_url: { посилання на API з даними про користувачів }  
token: { секретний ключ, необхідний для підключення }
```

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 50 |

Для інтеграції зі Slack, щоб користувачі могли оперативно отримувати сповіщення, необхідно налаштувати Slack-бот у воркспейсі компанії чи організації. Після його створення, стане доступним ключ, з яким додаток буде робити запити на сервера Slack, щоб відправляти повідомлення. Його схожим чином, як конфігурацію API, потрібно вписати у файл конфігурації в поле *token_bot* у секції *slack* файлу *config.yaml*.

Для інтеграції із Google Accounts, необхідно на сайті Google зареєструватись та згенерувати секретний і публічний ключі для доступу до API Google. Згенеровані дані необхідно записати у файл конфігурації *config.yaml* у секцію *oauth*.

Приклад налаштувань Google Accounts:

```
google_token_url: {посилання на API Google для генерування токена}  
google_userdata_url: {посилання на API Google для доступу до даних про користувача в Google}  
google_auth_url: {посилання на API Google для авторизації}  
google_secret: {секретний ключ Google}  
google_client_id: {публічний ключ Google}
```

4.2 Розгортання додатку

Для того, щоб розгорнути середовище для запуску розробленого додатку необхідно виконати наступні кроки:

- встановити та налаштувати PostgreSQL;
- встановити Python та його бібліотек;
- встановити ReactJS та його компоненти;
- встановити сучасний веб-браузер;
- налаштувати та запустити додаток.

4.2.1 Встановлення та налаштування PostgreSQL

Для встановлення бази даних PostgreSQL потрібно перейти на офіційний сайт PostgreSQL та завантажити 11 версію для необхідної платформи. Після встановлення, база даних буде готова для роботи та

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 51 |

доступна для підключення на порті, який визначить користувач, або на порті за замовчуванням.

Дані для підключення необхідно вказати в файлі конфігурації *config.yaml* у секції *db*.

4.2.2 Встановлення Python та його бібліотек

Для коректної роботи додатку, потрібно встановити Python версії 3. Для цього необхідно перейти на офіційний сайт Python та завантажити його. У проекті використовується Python версії 3.6, але можливе використання і більш ранніх версій, починаючи з 3.1.

Після встановлення Python, потрібно встановити фреймворки та бібліотеки, які використовуються. Повний перелік залежностей описано у файлі *requirements.txt*. Керування пакетами відбувається за допомогою пакетного менеджера PIP (*Python Package Manager*).

Щоб встановити всі необхідні пакети, достатньо використати команду *pip install -r requirements.txt* у головній папці додатку. Якщо встановлення відбулось успішно, то команда *pip freeze* виведе список всіх встановлених бібліотек.

4.2.3 Встановлення ReactJS та його компонентів

Фреймворк ReactJS працює на базі платформи Node.js. Мінімальної версією для стабільної роботи додатку є версія 8.2.4. Для встановлення Node.js потрібно перейти на офіційний сайт платформи та завантажити версію для необхідної платформи. Після завантаження, можна встановити сам React і його компоненти, для цього потрібно виконати команду *npm install*, яка встановить всі залежності, описані у файлі проекту *package.json*.

4.2.4 Встановлення сучасного веб-браузера

Для надійного функціонування та відображення клієнської частини додатку, необхідно взаємодіяти з нею за допомогою одного з сучасних веб-браузерів. Можна встановити Firefox версії 60+, Google Chrome версії 65+ та Safari версії 10+. Кожен з них можна встановити на відповідному офіційному сайті.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 52 |

4.2.5 Налаштування та запуск додатку

Налаштування проекту вказується у файлі конфігурації – *config.yaml*. Саме в ньому вказуються дані для доступу до бази даних, ключі для інтеграції з системами Slack, Google Accounts та API організації чи компанії. Після того, як файл конфігурації заповнено, можна запустити серверну частину веб-додатку. Це можна зробити виконавши команду *python main.py* у головній директорії проекту. Якщо запуск відбувся успішно, про це буде виведено відповідне повідомлення.

Після цього, потрібно запустити клієнтську частину, виконавши спочатку команду *npm run build*, щоб побудувати образ, а далі його запустити командою *npm start*.

Щоб переконатись, що додаток працює коректно, потрібно зайти у браузері за адресою <https://localhost:3000>. Якщо все працює коректно, то можна буде побачити головну сторінку сайту.

4.3 Розгортання додатку за допомогою Docker

За допомогою інструмента для управління ізольованими контейнерами Docker, можна значно спростити запуск та розгортання продукту. Головною перевагою є зберігання консистентності версій при запуску на будь-якому пристрої. Але, на жаль, на даний момент це можливо зробити лише на операційній системі Linux. Першим кроком, потрібно завантажити Docker, перейшовши на офіційний сайт продукту. Після встановлення, необхідно перейти до головної директорії проекту та побудувати контейнер командою *docker-compose build*, а далі запустити його, виконавши команду *docker-compose up*. В результаті чого, веб-додаток стане доступним за адресою <https://localhost:3000> в браузері.

Команди, які використовуються при виконанні команд *docker-compose build* та *docker-compose up*, можна побачити у файлах *Dockerfile.dev* та *docker-compose.yml* у головній директорії додатку. Вони дублюють функції, описані у пункті 4.2.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 53 |

Висновки до четвертого розділу

У цьому розділі було описано як можна інтегрувати розроблювальний додаток до вже існуючих систем, а також інструкцію до розгортання. В результаті чого можна зробити висновок, що додаток є простим в інтеграції до систем компаній чи організацій, а при наявності спеціальних інструментів, розгортання проекту є швидким та зручним.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| | | | | | | 54 |
| Зм. | Арк. | № докум. | Підпис | Дат | | |

ВИСНОВКИ

В даній дипломній роботі було розроблено та реалізовано оповіщувальну комунікаційну систему задля спрощення процесу привітань із днем народження членів організацій чи компаній. Програмний продукт було виконано на мові програмування Python, за допомогою веб-фреймворку aiohttp та із застосуванням сучасних технологій та методик для розробки, проектування та розгортання додатку.

У першому розділі роботи було проведено аналіз вже існуючих подібних систем, обґрунтовано створення нової системи, встановлено до неї вимоги, а також описано та проаналізовано предметну область дипломного проекту.

У другому розділі було проведено моделювання та конструювання оповіщувальної комунікаційної системи. Була розроблена архітектура додатка, проведено аналіз безпеки даних.

У третьому розділі дипломної роботи було описано методологію тестування програмного забезпечення. Було проаналізовано та протестовано систему, виконано контрольний приклад тестування.

У четвертому розділі було наведено інструкцію по інтегруванню оповіщувальної комунікаційної системи із вже існуючими додатками організацій чи компаній. Описано впровадження та розгортання застосунку на різних системах.

У рамках розробки даного дипломного проекту було застосовано знання з баз даних, клієнт-серверної архітектури програмного забезпечення, веб-розробки, безпеки даних, комп'ютерних мереж та використання методик і паттернів програмування.

Розроблений комплекс задач є повноцінним програмним продуктом, готовим для використання. Він може бути доповнений новими функціями та горизонтально масштабований.

| | | | | | | |
|-----|------|----------|--------|-----|--------------------|------|
| | | | | | ІАЛЦ.467100.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дат | | 55 |

ПЕРЕЛІК ПОСИЛАНЬ

1. Ramalho L. Clear, Concise, And Effective Programming / M. Blanchette, R. Roumeliotis. // Fluent Python. (March 2015). – P. 35 – 300.
2. Single-page application vs. multiple-page application [Електронний ресурс]. – 2016. – Режим доступу до ресурсу:
<http://medium.com/@NeotericEU/single-page-application-vs-multiple-pageapplication-2591588efe58>
3. Python <https://python.org> – офіційна сторінка у мережі Інтернет.
4. React <https://reactjs.org/> – офіційна сторінка у мережі Інтернет.
5. Docker <https://uk.wikipedia.org/wiki/Docker> – сторінка на Wikipedia.
6. Kubernetes <https://uk.wikipedia.org/wiki/Kubernetes> – сторінка на Wikipedia.
7. AioHTTP <https://github.com/aio-libs/aiohttp> – сторінка на GitHub.
8. HTTPS <https://uk.wikipedia.org/wiki/HTTPS> – сторінка на Wikipedia.
9. PIP [https://en.wikipedia.org/wiki/Pip_\(package_manager\)](https://en.wikipedia.org/wiki/Pip_(package_manager)) - сторінка на Wikipedia.

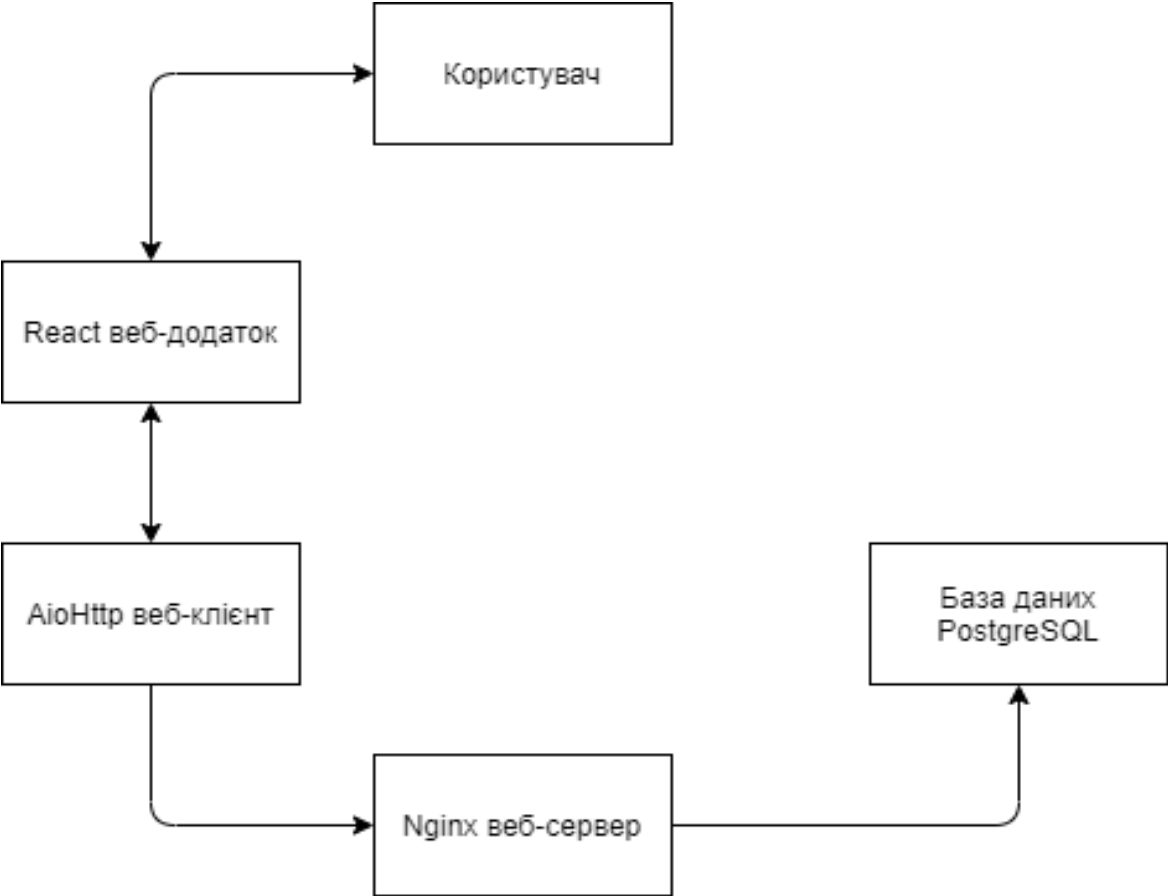
ДОДАТКИ

Схема алгоритму



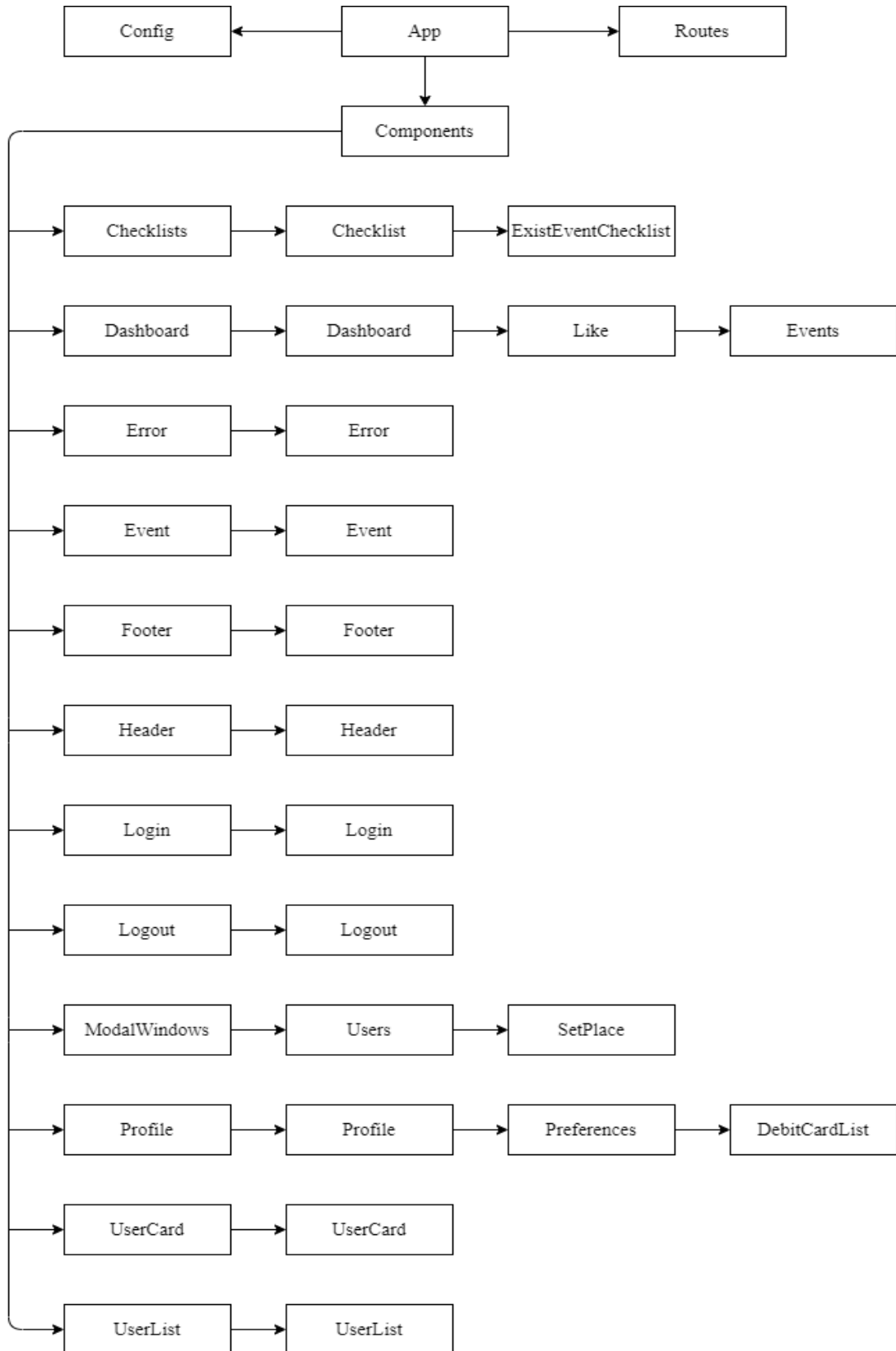
| | | | | | | | |
|-----------|------|-----------------|-------|-----|---|--------|----------|
| | | | | | ІАЛЦ.467100.002 Д1 | | |
| | | | | | | | |
| Зм. | Арк. | № докум. | Підп. | Дат | Оповіщувальна комунікаційна система Додаток А | | |
| Розробив | | Максимчук Д.А. | | | | | |
| Перевір. | | Виноградов Ю.М. | | | | | |
| Н. контр. | | Симоненко ВП. | | | | | |
| Затверд. | | | | | | | |
| | | | | | Літ. | Аркциш | Аркцишів |
| | | | | | | 1 | 1 |
| | | | | | НТУУ “КПІ імені Ігоря Сікорського”, ФІОТ Група ІО-64 | | |

Структурна схема



| | | | | | | | | | |
|-----------|------|-----------------|-------|-----|---|--|--|-------|---------|
| | | | | | ІАЛЦ.467100.002 Д2 | | | | |
| | | | | | | | | | |
| Зм. | Арк. | № докум. | Підп. | Дат | Оповіщувальна комунікаційна система Додаток Б | | Літ. | Аркуш | Аркушів |
| Розробив | | Максимчук Д.А. | | | | | | | |
| Перевір. | | Виноградов Ю.М. | | | | | | 1 | 1 |
| | | | | | | | НТУУ “КПІ імені Ігоря Сікорського”, ФІОТ Група ІО-64 | | |
| Н. контр. | | Симоненко ВП. | | | | | | | |
| Затверд. | | | | | | | | | |

Діаграма компонентів



| | | | | | | | | | | | | |
|-----------|------|-----------------|-------|-----|---|--|--|--|-------|---------|---|--|
| | | | | | ІАЛЦ.467100.002 ДЗ | | | | | | | |
| Зм. | Арк. | № докум. | Підп. | Дат | Оповіщувальна комунікаційна система Додаток В | | | Лім. | Аркцш | Аркцшів | | |
| Розробив | | Максимчук Д.А. | | | | | | | | 1 | 1 | |
| Перевір. | | Виноградов Ю.М. | | | | | | НТУУ “КПІ імені Ігоря Сікорського”, ФІОТ Група ІО-64 | | | | |
| | | | | | | | | | | | | |
| Н. контр. | | Симоненко ВП. | | | | | | | | | | |
| Затверд. | | | | | | | | | | | | |